**DOSSIER DE PROJET**

**Lionel ENSFELDER**

***Session 1 – Année 2019***

****

**Développeur web et web mobile**

***Titre professionnel de niveau 5***

SOMMAIRE

Table des matières

[Chapitre 1 – Conception d’une application web ou web mobile 1](#_Toc24876276)

[1. Liste des compétences du référentiel couvertes par le projet 1](#_Toc24876277)

[2. Résumé du projet 1](#_Toc24876278)

[3. Cahier des charges, expression des besoins ou spécifications fonctionnelles du projet 2](#_Toc24876279)

[4. Spécifications techniques du projet 9](#_Toc24876280)

# Chapitre 1 – Conception d’une application web ou web mobile

## Liste des compétences du référentiel couvertes par le projet

**Les compétences du référentiel couvertes par le projet sont :**

1. **Développer la partie Front-End d’une application web ou web mobile en intégrant les recommandations de sécurité :**

* Maquetter une application
* Réaliser une interface utilisateur web statique et adaptable
* Développer une interface utilisateur web dynamique

1. **Développer la partie Back-End d’une application web ou web mobile en intégrant les recommandations de sécurité :**

* Créer une base de données
* Développer les composants d’accès aux données
* Développer la partie Back-End d’une application web ou web mobile

## Résumé du projet

Movies Manager est une interface web de gestion de films qui se base sur les données rentrées par l’utilisateur. Cette application est destinée dans un premier temps à rentrer les données d'un film manuellement via un formulaire puis les stocker en base de données. La restitution de tous les films présents en base de données et des informations associées se fait via la page d'accueil. Tous les utilisateurs connectés ont la possibilité de modifier les informations de tous les films présents en base de données. Ils pourront ainsi entrer et mettre à jour pour chaque film : le titre, l’année, le tag associé, le(s) acteur(s), le réalisateur et enfin la jaquette.

Pour ce projet j’ai choisi d’utiliser HTML, CSS, Bootstrap, Twig et JavaScript pour la partie Front-End. La partie Back-End elle, est construite autour du stack Apache, PHP, MySQL (uWamp sur clé USB) et comporte aussi du JavaScript.

J’ai choisi de ne pas utiliser de Framework Back-End pour m’exercer au PHP et travailler la Programmation orientée objet (POO). En effet, la POO est utilisée par de nombreux Framework et est une façon de programmer abstraite, flexible et très populaire. Ce projet me permettra à terme de comprendre plus facilement les choix d'implémentation lors de l'abord d’un nouveau Framework et d’ajouter un niveau d’abstraction à mon style de programmation.

## Cahier des charges, expression des besoins ou spécifications fonctionnelles du projet

1. **Contexte**

En tant que geek passionné de technologies et gros consommateur de films et de séries j’ai depuis longtemps envie d’avoir un serveur domestique sur lequel je ferai tourner des applications web accessibles par l’intermédiaire d’un Dashboard.

Ce tableau de bord serait matérialisé par une tablette me permettant d’avoir facilement accès à diverses informations comme la météo, mon calendrier, mes mails, l’état du trafic routier, mes playlists Spotify, ou encore une liste de films à voir.

Cette application web est donc un exercice préparatoire à la réalisation de ce projet personnel tout en satisfaisant aux besoins de la certification.

1. **Cible**

La cible principale de cette application est moi-même ainsi que les personnes présentes à mon domicile ayant un intérêt pour les médias en particulier les films. Ces personnes sont désireuses de garder à porter de main des informations sur des films à voir et pouvoir les modifier en cas de besoin. La simplicité de l’interface et un choix réduit de fonctionnalités sont très importants car la cible doit s’approprier l’application rapidement.

1. **Objectifs**

**La réalisation de cette application web à plusieurs objectifs :**

* Permettre la gestion (ajout, modification et suppression) d’une bibliothèque d’informations à propos de films.
* Mettre en œuvre la persistance des données (C.R.U.D) dans une application web.
* Partir d’une implémentation en programmation fonctionnelle et refactoriser le code en programmation orienté objet.

1. **Spécifications fonctionnelles des besoins – périmètre fonctionnel**
2. **User stories**

**En tant que [utilisateur anonyme],** je souhaite [créer un compte] afin d’[avoir accès aux fonctionnalités du site].

**En tant que [utilisateur anonyme],** je souhaite [me connecter à mon compte] afin d’[avoir accès aux fonctionnalités du site].

**En tant que [utilisateur connecté],** je souhaite [voir tous les films enregistrés an base de données] afin de [sélectionner un film dans cette liste].

**En tant que [utilisateur connecté],** je souhaite [ajouter un film dans la liste des films] afin de [le retrouver plus tard].

**En tant que [utilisateur connecté],** je souhaite [modifier les informations d’un film] afin de [rectifier une erreur].

**En tant que [utilisateur connecté],** je souhaite [supprimer un film de la liste des films] afin de [ne plus le voir apparaitre dans la liste des films].

**En tant que [utilisateur connecté],** je souhaite [voir les informations de mon compte utilisateur] afin de [vérifier leurs exactitudes].

**En tant que [utilisateur connecté],** je souhaite [modifier les informations de mon compte] afin de [rectifier une erreur ou changer d’avis].

**En tant que [utilisateur connecté],** je souhaite [me déconnecter] afin de [verrouiller les fonctionnalités du site].

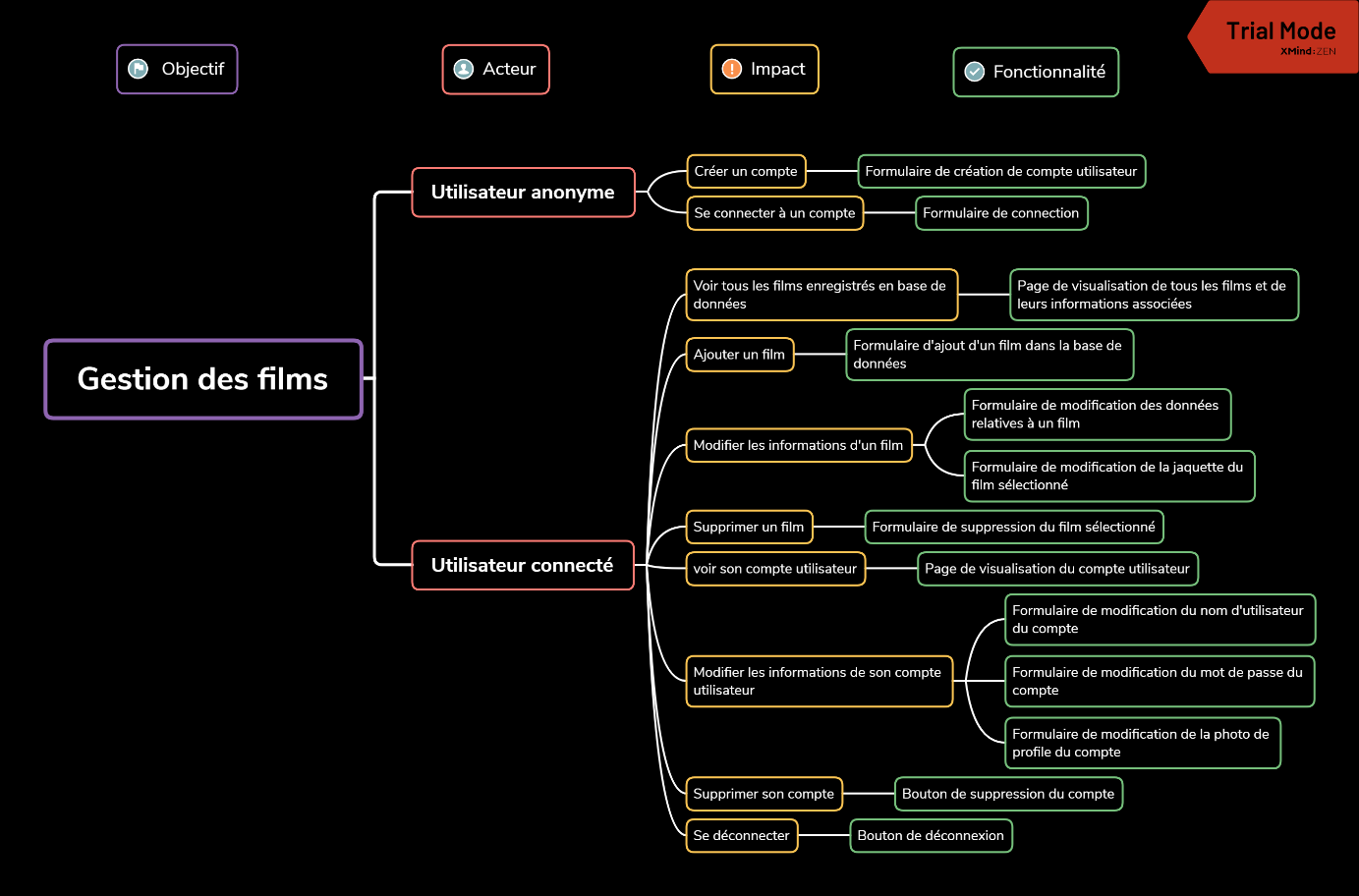
**En tant que [utilisateur connecté],** je souhaite [supprimer mon compte] afin de [ne plus revenir sur ce site].

1. **Impact mapping du projet**

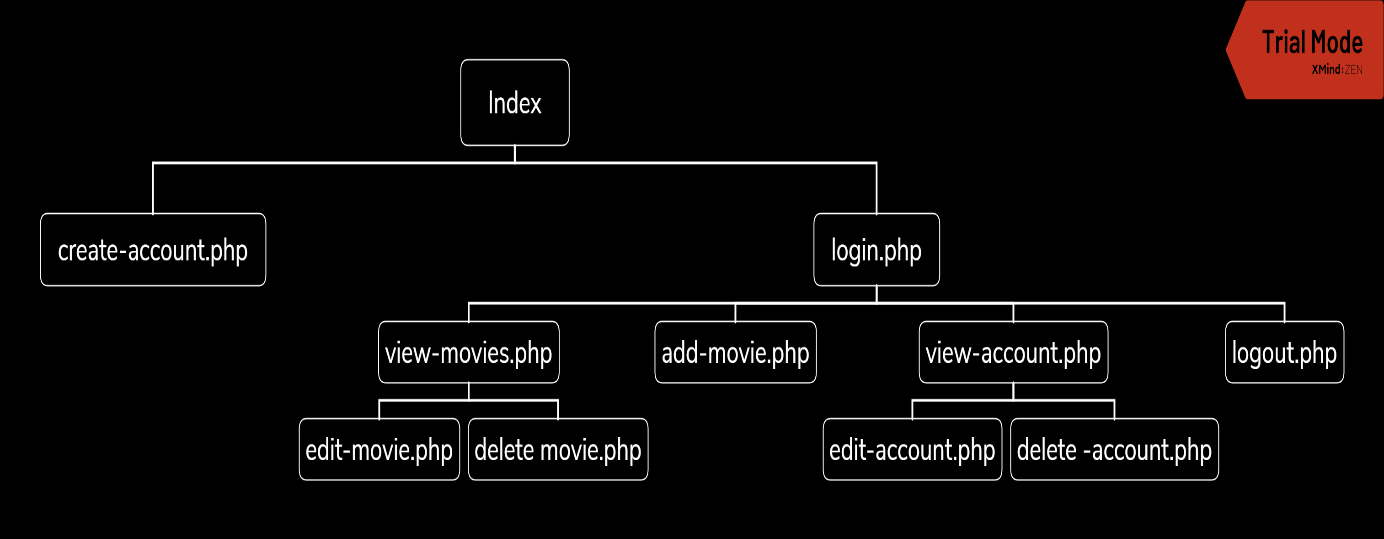
Pour ce projet relativement simple nous j’utilise l’impact mapping plutôt qu’un diagramme UML. L’impact mapping à l’avantage d’avoir une mise en œuvre simple et d’être très clair sur le périmètre fonctionnel d’un projet.

L’impact mapping est une technique permettant de représenter sous forme d’un graphique les fondamentaux d’un projet. **Il a pour but de se concentrer sur l’essentiel pour ne pas se disperser et répond à 4 grandes questions :**

* **Pourquoi ?** \*Impact. Le fondement même du projet. Le(s) besoin(s) auquel(s) il répond.
* **Qui ?** \*Acteur. Les acteurs, utilisateurs, personnes impactées par le projet.
* **Comment ?** \*impact. Les comportements qui vont permettre d’atteindre l’objectif.
* **Avec quelle(s) fonctionnalité(s) ?** \*Fonctionnalité. Le(s) fonctionnalité(s) permettant de créer le comportement pour l’impact souhaité.



1. **Structure des pages**



1. **Responsive**

L’application sera conçue pour s’adapter aux principales contraintes en matière de taille d’écran et de types d’appareils.

**Elle devra s’adapter aux appareils suivants pour une navigation optimale :**

* Téléphone portable (de px à px)
* Tablette (de px à px)
* Ordinateur portable (de px à px)
* Ordinateur de bureau (de px à px)

La barre de navigation devra s’adapter en fonction de la largeur et du type d’appareil. Une attention particulière sera faite à l’adaptation sur mobile.

1. **Accessibilité**

Bien que l’application n’ait pas vocation à être disponible sur internet le respect de quelques recommandations au sujet de l’accessibilité est requis :

* Proposer des équivalents textuels à tout contenu non textuel.
* Rendre la navigation au clavier possible dans toute l’application.
* Ne pas utiliser de contenu susceptible de provoquer des crises convulsives.
* Rendre les contenus textuels lisibles et compréhensibles en utilisant une police sans empattement.

1. **Contraintes des principales fonctionnalités**

|  |  |
| --- | --- |
| **Fonctionnalité** | **Contrainte(s) associée(s)** |
| **Créer un compte utilisateur** | Pas de doublons de noms d’utilisateurs + vérifier les données en entrée. |
| **Se connecter à un compte utilisateur** | Impossible si utilisateur connecté. |
| **Voir la liste de tous les films** | Utilisateur connecté. |
| **Ajouter un film** | Vérifier les données en entrée. |
| **Modifier les informations d’un film** | Vérifier les données en entrée + renommer la jaquette associée. |
| **Modifier la jaquette d’un film** | Extensions autorisées uniquement. |
| **Supprimer un film** | Supprimer la jaquette associée. |
| **Voir les informations de son compte** | Session active. |
| **Modifier le nom d’utilisateur** | Session active + vérifier le mot de passe. |
| **Modifier le mot de passe** | Session active + vérifier les données en entrée + vérifier le mot de passe actuel. |
| **Modifier la photo de profile** | Extensions autorisées uniquement + renommer image. |
| **Se déconnecter** | Suppression de toutes les sessions. |

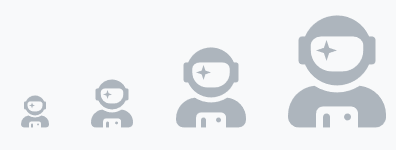
1. **Langue**

La langue de l’application est l’anglais.

1. **Charte graphique**

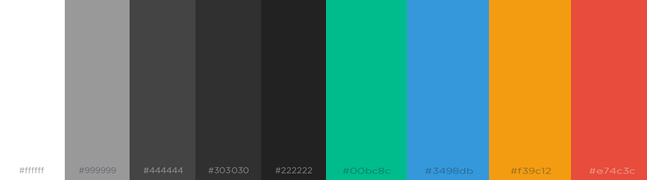
**Nom :** Movies Space

**Logo :**



**Favicon :**

**Palette de couleurs :**

****

**Typographie :** Montserrat (<https://fonts.google.com/specimen/Montserrat>)



**Icones :** la partie gratuite de la police d’icônes [FontAwesome](https://fontawesome.com/) devrait suffire pour les besoins de ce projet.

**Type de design :** Flat design

« Le Flat Design est un style de design web excluant tous détails superflus de la création. Il se caractérise par des aplats de couleurs sobres mais actuelles et dont l’organisation de la typographie créée la structure du contenu. Le Flat design est un style minimaliste, extrêmement clair et épuré. « Less is more » prend ici tout son sens. olybop.fr »

**Thème :** l’interface sera de type « dark mode » pour coller au thème des films et du cinéma. L’utilisateur sera en mesure de naviguer dans l’application sans gêne visuelle ou éblouissement. Aucun thème « light » n’est pour l’instant prévu.

## Spécifications techniques du projet

1. **Gestion de projet et organisation**

Pour la gestion du projet j’ai choisi d’utiliser le logiciel Trello, la méthodologie Kanban et en partie la méthodologie Agile Scrum (user story, backlog, sprint).

**La méthode Kanban**

Cette méthode de gestion de projet a été inventé par Taiichi Ōno pour Toyota en 1950. Son but est d’optimiser la production à flux tendu. Dans la mesure où je suis seul sur ce projet j’ai utilisé le système de cartes de la méthodologie Kanban pour me focaliser sur l’action (limiter les tâches parallèles par exemple) et savoir à tout moment ou je me situe concrètement dans la réalisation.

**La méthodologie Agile**

La méthodologie Agile est basée sur un cycle de développement qui place le client et les fonctionnalités au centre de l’équation. Quand cette méthodologie est en place de client / demandeur à une meilleur visibilité de l’avancée des travaux. Il peut ainsi faire un feedback rapide sur les fonctionnalités du projet. Cette mise en valeur du client est permise grâce au fonctionnement de la méthodologie qui est simple : proposer une version minimale du produit puis intégrer des fonctionnalités supplémentaires par processus itératifs.

Source : https://fr.wikipedia.org/wiki/M%C3%A9thode\_agile.

**Scrum**

Scrum est une méthode Agile dédiée à la gestion de projet. Ce Framework a pour objectif d’améliorer la productivité de son équipe.

“Scrum, c’est comme le poker, vous apprenez les règles en 10 minutes, mais vous mettez du temps pour devenir vraiment bon. David Matthew ».

Source : https://fr.wikipedia.org/wiki/Scrum\_(d%C3%A9veloppement)

**Trello**

Trello est un outil collaboratif conçu pour organiser ses tâches et gérer ses projets. L’outil est sous forme d’application web ou d’application native. Pour commencer il faut juste créer un tableau de bord thématique (ici le nom du projet) et créer des colonnes, des cartes ainsi que des tags.

Les colonnes vont permettre d’organiser les tâches concrètes après découpage. **J’ai opté pour une organisation en 4 colonnes :**

* **Product Backlog :** ce que je dois implémenter, découper en très petites tâches. Cette liste comprend aussi mes idées, listes, tâches, pense-bête en vrac.
* **Sprint Planning :** Les tâches du sprint actuel (j’ai décidé qu’un sprint est égal à une semaine pour ce projet)
* **In Progress :** ce que je fais en ce moment. Pas plus de 2 cartes du sprint en cours en même temps dans cette colonne.
* **Done :** les tâches faites sont transférées dans cette colonne.

Chaque carte représente une tâche ou une micro tâche et comporte un ou plusieurs tags ainsi que des « stories points ». Les stories points sont des mesures chiffrées destiné à estimer l’effort à fournir pour implémenter une fonctionnalité ou réaliser une tâche. Plus la tâche est complexe et plus la valeur du « story point » est grande.

Les tags permettent de regrouper les cartes par thème / domaine. **J’ai choisi d’implémenter les tags par thème :**

* **Movies :** tout ce qui concerne la gestion ou l’affichage d’un/des film(s).
* **Profile / user :** tout ce qui concerne la gestion ou l’affichage d’un élément concernant l’utilisateur.
* **Bug :** les bugs.
* **Security :** la sécurité.
* **Documentation / writing :** la documentation du code et la rédaction du projet pour la certification.
* **Design / responsive / accessibility :** tout ce qui concerne le design, le design adaptable et l’accessibilité de l’interface.
* **BDD :** script de gestion et actions directes sur la base de données.
* **Test :** tests unitaires.
* **Cleaning / optimisation :** nettoyage et optimisation du code en vue de sa présentation, sa relecture et sa maintenance.

1. **Maquettage**

Le maquettage est réalisé avec le logiciel Figma ainsi que la librairie de composants Bootstrap de Figma.

1. **Supports / navigateurs**

L’interface et les fonctionnalités devront être compatible avec les navigateurs récents supportant CSS Flexbox et Grid :

* Edge
* Chrome
* Firefox
* Safari
* Opéra

1. **Versionning**

Le versionning du projet est assuré par git et github.

**Github a été choisi pour les raisons suivantes :**

* Plateforme populaire et très bien documentée.
* Les besoins du projet sont basiques.
* Je travaille déjà avec cette plateforme et donc elle s’intègre parfaitement à mon workflow.

La branche master sera la branche principale durant le développement de la première version qui sera implémentée en programmation fonctionnelle. Au fur et à mesure des refactorisations une branche sera créée pour chaque changement majeur.

Une intégration du moteur de templates Twig fera également l’objet d’une branche distincte.

1. **Structure des fichiers**
2. **Type de contenu**

Cette application web est destinée à présenter sous forme structurée les informations détaillées relative à une liste de films. Le contenu géré ici est essentiellement du texte ainsi que des images.

Tout le contenu modifiable du site provient d’entrées utilisateur. Lors de l’ajout ou de la modification d’un film les images acceptés sont pour l’instant uniquement au format image/jpeg et les extensions acceptées sont .jpg et .jpeg.

1. **SEO**

**Bien que l’application soit uniquement locale une partie des bonnes pratiques SEO sera respecté pour les besoins de la certification :**

* Nommer les jaquettes de films de façon explicite
* Remplir la balise TITLE et ALT de chaque image avec son contexte
* Hiérarchiser le contenu de l’application en suivant les bonnes pratiques sémantiques
* Remplir correctement les données META pour chaque page (title, description, keywords, robots)

1. **Front-End**

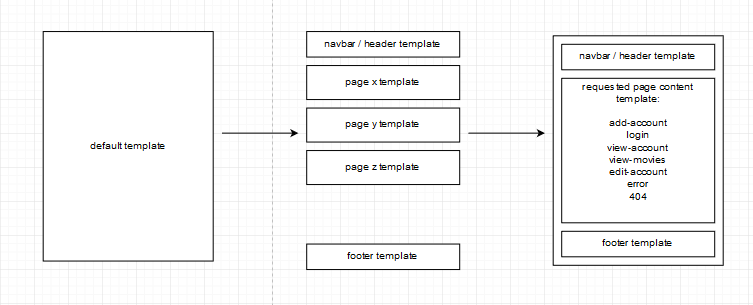
**HTML5 et CSS3 :**

* HTML5 (HyperText Markup Language 5) pour structurer les pages.
* CSS3 (Cascading Style Sheets) pour formater les pages.

**Bootstrap :** le Framework Bootstrap dans sa version 4 sera utilisé comme principal support à la conception de l’interface de l’application. L’interface sera agencée selon les bonnes pratique d’organisation en grille via les outils fournis par Bootstrap (.container, container-fluid, row, col etc…).

**Boostwatch :** afin de coller au design sur fond noir souhaité sans augmenter le temps d’implémentation de l’interface un thème fournit par Bootswatch ([Darkly theme](https://bootswatch.com/darkly/)) sera utilisé. Celui-ci est spécialement conçu pour surcharger le CSS de base du Framework Bootstrap.

**Twig :** Twig est un moteur de templates pour le langage de programmation PHP, utilisé par défaut par le Framework Symfony. Il sera utilisé pour permettre de créer des composants graphiques réutilisables et flexibles. La fonctionnalité d’héritage de template servira ici à imbriquer des composants (qui sont aussi des templates) tel qu’une barre de navigation ou un pied de page à un modèle général qui sera la base de toutes les pages.



|  |  |
| --- | --- |
| **Nom du composant** | **Forme du composant** |
| Ex : formulaire édition d’un film | Ex : modal + form bootstrap |

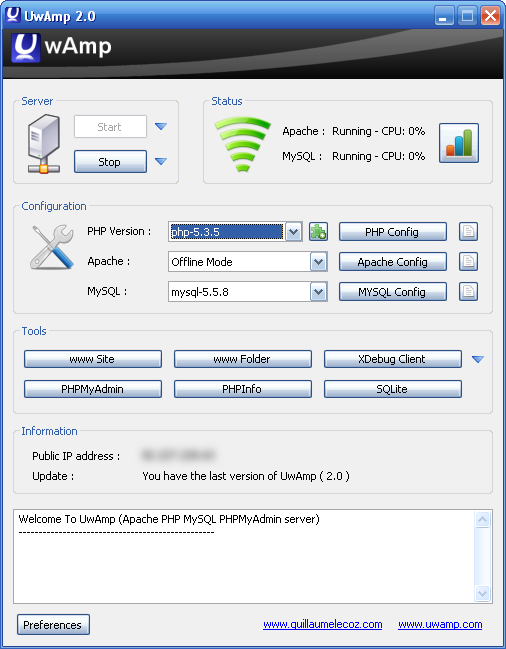
**Javascript :** JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives (exécution coté client) mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js.

Certains composants graphiques de l’application requièrent javascript pour fonctionner correctement et fournir à l’utilisateur une meilleure expérience. **Les composants concernés sont :**

* Les alertes
* Les boutons
* Les fenêtres modales
* La barre de navigation (responsive)

1. **Back-End**
2. **Serveur Web**

**uWamp :** uWamp est un serveur Wamp Apache MySQL PHP et SQLite. Il est destiné à être lancer depuis une clé USB ce qui le rend facilement portable. Cette application est parfaitement adaptée à ce projet car le temps de travail sera partagé entre le centre AFPA, le domicile et éventuellement le lieu du stage si besoin.

****

1. **Responsive / web mobile**
2. **Sécurité**
3. **Gestion des erreurs**
4. **Accessibilité**
5. **Organisation des données et base de données**
6. **Documentation**